33장. 7번째 데이터 타입 Symbol

33.1 심벌이란?

1. 자바스크립트에는 6개의 타입 즉, 문자열, 숫자, 불리언, undefined, null, 객체 타입이 있었다.

2. 심벌(Symbol)은 ES6에서 도입된 7번째 데이터 타입으로, 변경 불가능한 원시 타입의 값.

   ➡️ 주로 이름의 충돌 위험이 없는 유일한 프로퍼티 키를 만들기 위해 사용

33.2 심벌 값의 생성

33.2.1 Symbol 함수

1. 심벌 값은 Symbol 함수를 호출하여 생성, 다른 값과 절대 중복되지 않는 유일무이한 값이다.
   - new 연산자와 함께 호출 X

2. Symbol 함수dpsms 선택적으로 문자열을 인수로 전달할 수 있다.
   - 값에 대한 설명으로 디버깅 용도로만 사용

3. 심벌 값도 객체처럼 접근하면 암묵적으로 래퍼 객체를 생성
   - 암묵적으로 문자열, 숫자 타입으로 변환 X
   - 단, 불리언 타입으로는 암묵적으로 타입 변환된다.

33.2.2 Symbol.for / Symbol.keyFor 메서드

1. **Symbol.for 메서드**

   : 인수로 전달받은 문자열을 키로 사용하여, 키와 심벌 값의 쌍들이 저장된 전역 심벌 레지스트리에서 해당 키와 일치하는 심벌 값을 검색

   - 검색 성공 → 새로운 심벌 값 생성 X, 검색된 심벌 값을 반환
   - 검색 실패 → 새로운 심벌 값 생성, 인수로 전달된 키로 전역 심벌 레지스트리에 저장 후, 생성된 심벌 값을 반환

2. **Symbol.keyFor 메서드**를 사용하면 전역 심벌 레지스트리에 저장된 심벌 값의 키를 추출 가능

33.3 심벌과 상수

1. 위, 아래, 왼, 오를 나타내는 상수를 정의한다고 생각

```
// 위, 아래, 왼, 오를 나타내는 상수를 정의
// 중복될 가능성이 없는 심벌 값으로 상수 값을 생성
const Direction = {
    UP : Symbol('up'),
    DOWN : Symbol('down'),
    LEFT : Symbol('left'),
    RIGHT : Symbol('right')
};

const myDirection = Direction.UP;

if ( myDirection === Direction.UP ){
    console.log('U are going UP.');
}
```

💡 **enum**

1. enum은 명명된 숫자 상수의 집합으로 열거형이라고 부른다.
2. 자바스크립트에서는 지원X, 타입스크립트에서는 enum을 지원한다.
3. 자바스크립트에서 흉내내어 사용하려면 Object.freeze 메서드와 심벌 값을 사용한다.

33.4 심벌과 프로퍼티 키

1. 객체 프로퍼티 키는 모든 문자열 또는 심벌 값으로 만들 수 있으며, 동적으로 생성 가능

2. 심벌 값을 프로퍼티 키로 사용하려면 프로퍼티 키로 사용할 심벌 값에 대괄호 사용해야 함.

   프로퍼티에 접근할 때도 마찬가지로 대괄호 사용.

```
const obj = {
    // 심벌 값으로 프로퍼티 키를 생성
    [Symbol.for('mySymbol')] : 1
```

```
};

obj[Symbol.for('mySymbol')]; // 1
```

3. **심벌 값은 유일무이한 값이므로 심벌 값으로 프로퍼티 키를 만들면,**
   다른 프로퍼티 키와 절대 충돌하지 않는다.

33.5 심벌과 프로퍼티 은닉

1. 심벌 값을 프로퍼티 키로 사용하여 생성한 프로퍼티는,
   for ... in 문, Object.keys, Object.getOwnPropertyNames 메서드로 찾을 수 없다.

2. 완전히 숨길 수 있는 것은 X
   ES6에서 도입된 Object.getOwnPropertySymbols. 메서드 사용하면 찾을 수 있다.

33.6 심벌과 표준 필트인 객체 확장

1. 일반적으로 표준 빌트인 객체에 사용자 정의 메서드 직접 추가하는 것은 권장 X
   표준 빌트인 객체는 읽기 전용으로 사용하는 것이 좋음.

2. 그 이유는 개발자가 직접 추가한 메서드와 미래에 표준 사양으로 추가될 메서드 이름이
   중복될 수 있기 때문이다.
   ▶ 심벌을 이용하면 충돌, 중복 위험 X 안전하게 확장 가능

33.7 Well-known Symbol

1. 자바스크립트가 기본 제공하는 빌트인 심벌 값을 ECMA Script 사양에서는
   Well-known Symbol 이라 부른다. 이는 JS 엔진 내부 알고리즘에 사용된다.

2. 만약 빌트인 이터러블이 아닌 일반 객체를 이터러블처럼 동작하도록 구현하고 싶다면,
   이터레이션 프로토콜을 따르면 된다.
   즉, **Well-known Symbol**인 Symbol.iterator를 키로 갖는 메서드를 객체에 추가하고,
   이터레이터를 반환하도록 구현하면 그 객체는 이터러블이 된다.

3. 심벌은 중복되지 않는 상수 값을 생성하는 것은 물론,
   기존 작성된 코드에 영향을 주지 않고 새로운 프로퍼티를 추가하기 위해,
   즉, **하위 호환성**을 보장하기 위해 도입되었다.